REMARKS

In response to the Office Action mailed January 21, 2005, Applicant respectfully requests reconsideration. To further the prosecution of this Application, Applicant submits the following remarks and has added new claims. The claims as now presented are believed to be in allowable condition.

Claims 1-30 are pending in this Application. Claims 1, 8, 15 and 22 are independent claims.

Preliminary Matters

The Advisory Action mailed May 9, 2005 states that the Patent Office withdraws the rejection of claim 1 under 35 USC §103(a) as being unpatentable over Applicant's Admitted Prior Art (AAPA) in view of U.S. Patent No. 5,278,974 (Lemmon et al.) and in further view of U.S. Patent No. 6,237,077 (Sharangpani et al.). The Advisory Action further states that this rejection was incorporated into claim 2.

Upon inquiry by Applicant's Representative, David E. Huang, on May 19, 2005, the Examiner clarified that the Advisory Action contained a typographical error. In particular, the Examiner indicated that claim 2 is not rejected under 35 USC §103(a) as being unpatentable over AAPA in view of Lemmon and in further view of Sharangpani et al., but that claim 23 is so rejected.

Objection to the Drawings

The drawings were objected to because they included a reference character not mentioned in the description. Applicant has amended the description to cure this minor informality. In particular, one of ordinary skill in the art would know that the Applicant's disclosed payload field 126 included a series of results 126-1, 126-2, 126-3 as illustrated in Fig. 6, and Applicant has amended the description to reflect this fact. No new matter has been added. Accordingly, the objection to the drawings should be withdrawn.

Rejections under §102 and §103

Claims 1, 4-5 and 7 were rejected under 35 U.S.C. §102(e) as being anticipated by Sharangpani et al. Claim 2 was rejected under 35 U.S.C. §103(a) as being unpatentable over Sharangpani et al. in view of Hennessy and Patterson, Computer Architecture – A Quantitative Approach, 2nd Edition, 1996 (Hennessey). Claim 3 was rejected under 35 U.S.C. §103(a) as being unpatentable over Sharangpani et al. in view of U.S. Patent No. 5,748,978 (Narayan et al.). Claim 6 was rejected under 35 U.S.C. §103(a) as being unpatentable over Sharangpani et al. in view of U.S. Patent No. 6,233,671 (Abdallah et al.). Claims 8, 11-12, 14-15, 18-19 and 21-22 were rejected under 35 U.S.C. §103(a) as being unpatentable over (AAPA) in view of Lemmon et al. in further view of Sharangpani et al. Claims 9 and 16 were rejected under 35 U.S.C. §103(a) as being unpatentable over AAPA in view of Lemmon in view of Sharangpani et al., and in further view of Hennessey. Claims 10 and 17 were rejected under 35 U.S.C. §103(a) as being unpatentable over AAPA in view of Lemmon in view of Sharangpani et al., and in further view of Narayan et al. Claims 13 and 20 were rejected under 35 U.S.C. §103(a) as being unpatentable over AAPA in view of Lemmon in view of Sharangpani et al., and in further view of Abdallah et al.

Based on Applicant's inquiry, claims 23-30 were further rejected under 35 USC §103(a) as being unpatentable over AAPA in view of Lemmon et al. and in further view Sharangpani et al. Claims 23-30 were rejected under 35 USC §103(a) as being unpatentable over AAPA in view of Lemmon et al. and in further view Sharangpani et al.

Applicant respectfully traverses each of these rejections and requests reconsideration. The claims are in allowable condition because they patentably distinguish over the cited references.

Sharangpani explains modem processors have the capacity to process multiple instructions concurrently at very high rates (column 1, lines 11-13), and that there has been increasing focus on methods to identify and exploit the

instruction level parallelism ("ILP") needed to fully utilize the capabilities of modem processors (column 1, lines 30-33). Along these lines, Sharangpani discloses a method of processing branch instructions efficiently (column 1, lines 66-67). In particular, one or more branch instructions are placed in an instruction bundle with the instructions ordered in an execution sequence within the bundle (column 2, lines 4-8). The bundles of instructions are transferred to execution units indicated by a template field that is associated with the bundle (column 2, lines 8-10). The first branch instruction in the bundle's execution sequence that is resolved taken is determined, and retirement of subsequent instructions in the execution sequence is suppressed (column 2, lines 10-13). In accordance with a 64-bit implementation, an instruction bundle 200 includes three instruction fields or slots 210(a)-210(c), a template field 220, and a stop field 230 (column 4, lines 54-60 and Fig. 2). The template field 220 encodes a template type that indicates how instruction slots 210 are mapped to execution units (column 4, lines 64-67). Such operation enables processing of instruction groups and suppression of instructions subsequent to taken branches (column 5, lines 57-62).

Applicant's Admitted Prior Art (AAPA) discloses a conventional data storage system having memory circuit boards which are configured to respond to a small set of basic instructions such as READ, WRITE, ADD and MASK-COMPARE-AND-SWAP commands (page 2, lines 5-7). Unfortunately, communications between interfaces and the memory circuit boards of the data storage system require several separate communications (page 2, lines 24-25). The large number of communications results in poor response time due to delays, e.g., handshaking delays in waiting for a memory circuit board to complete an operation incrementally before sending a communication for a next operation, resource contention delays due to contention for a bus or buses, etc. (page 3, lines 3-8).

Lemmon discloses that the bandwidth of a first bus and a second bus may be unequal due to differences in protocol overheads and cycle times between the buses (Abstract, lines 1-3). Lemmon discloses equalizing the bandwidth

between the buses without sacrificing any bandwidth on the lower bandwidth bus and without introducing any buffering in a control logic device (Abstract, lines 3-6). The control logic equalizes the bandwidths of the buses by instructing a device coupled to the second bus to insert a partial dead bus cycle in a read transmission thereby adjusting read timing on the second bus when the second bus is heavily loaded (Abstract, lines 6-11).

Claims 1-7 and 23-24

Claim 1 is directed to a method which is performed in a memory circuit board that stores a cache for a data storage system. The method includes the step of receiving a communication that includes a script command and a payload. The payload includes a series of individual instructions. The method further includes the step of parsing the payload to identify the series of individual instructions in response to the script command. The method further includes the step of performing a series of operations in accordance with the identified series of individual instructions.

The Sharangpani does not disclose a method having the steps of parsing a payload to identify a series of individual instructions in response to a script command, and performing a series of operations in accordance with the identified series of individual instructions, as recited in claim 1. Rather, in Sharangpani which deals with instruction level parallelism (e.g., see column 1, lines 30-33 of Sharangpani), the first branch instruction in a bundle's execution sequence that is resolved taken is determined, and retirement of other branch instructions in the execution sequence is suppressed (e.g., column 2, lines 10-13 of Sharangpani). Accordingly, Sharangpani does not disclose parsing a payload to identify a series of individual instructions in response to a script command, and performing a series of operations in accordance with the identified series of individual instructions as recited in claim 1.

In the §102(e) rejection of claim 1, the Office Action contends at the bottom of page 3 that Sharangpani discloses, in Table 2 in column 6, parsing a

payload to identify a series of individual instructions. Applicant respectfully disagrees with this contention. Table 2 in column 6 does **not** teach parsing a payload to identify a series of individual instructions, as recited in claim 1. Rather, with reference to Table 2 as well as Fig. 2, the Sharangpani template field 220 encodes the position of any instruction group boundaries within instruction bundle 200, as well as template type that indicates how instruction slots are mapped to execution units (see Column 4, lines 64-67). Even the Office Action states at the bottom of page 3 that, based on a template of Table 2, the instructions and their types are determined and then are routed to execution units.

The Office Action further contends at the top of page 4 that Sharangpani discloses, in the Table in column 5, performing a series of operations in accordance with an identified series of individual instructions. Again, Applicant respectfully disagrees with this contention. The Table in column 5 of Sharangpani does **not** disclose performing a series of operations in accordance with an identified series of individual instructions, as recited in claim 1. Rather, the Table is simply a list of widely supported instruction types that are also supported in the 64-bit ISA (see column 5, lines 21-23 of Sharangpani). Accordingly, even if one were to argue that Sharangpani discloses parsing a payload to identify a series of individual instructions in response to a script command, Sharangpani still does not disclose performing a series of operations in accordance with the identified series of individual instructions, as recited in claim 1.

Clearly, the standard and irrebuttable meaning of the word "series" is "a group of things (e.g., operations) of the same class coming one after the other in succession" (emphasis added); see The American Heritage Dictionary, Second Edition, Houghton Mifflin Company, Boston, 1991). Accordingly, performing a series of operations in accordance with an identified series of individual instructions clearly means performing operations one after the other in succession (based on the series of individual instructions parse from the payload

of the script command communication). In contrast, Sharangpani exploits instruction level parallelism ("ILP") needed to fully utilize the capabilities of modem processors (e.g., column 1, lines 30-33) by routing the slotted instructions within an instruction bundle 200 to different execution units (e.g., column 4, lines 64-67).

Moreover, it is unclear how one could modify the Sharangpani method or why one would want to modify the Sharangpani method to perform a series of operations in accordance with an identified series of individual instructions as recited in claim 1 since Sharangpani purposefully wants to suppress instructions subsequent to taken branches (e.g., column 5, lines 57-62 of Sharangpani). Nothing in any of the other references (e.g., AAPA, Lemmon, etc.) discloses how one could successfully make such a modification. **If anything, it seems that serializing the operations in Sharangpani would defeat the purpose of Sharangpani.**

For the reasons stated above, claim 1 patentably distinguishes over Sharangpani, and the rejection of claim 1 under 35 U.S.C. §102(e) should be withdrawn.

Because claims 2-7 and 23-24 depend from and further limit claim 1, claims 2-7 and 23-24 patentably distinguish over Sharangpani for at least the same reasons.

Claims 8-14 and 25-26

Claim 8 is directed to a data storage system which includes (a) a set of storage devices; (b) a memory circuit board that stores a cache to temporarily store copies of data elements stored in the set of storage devices; and (c) a processor circuit board that operates as at least one of a front-end interface between an external device and the cache and a back-end interface between the cache and the set of storage devices. The memory circuit board is configured to (i) receive, from the processor circuit board, a communication that includes a script command and a payload, the payload including a series of individual

instructions, (ii) parse the payload to identify the series of individual instructions in response to the script command, and (iii) perform a series of operations in accordance with the identified series of individual instructions.

The cited prior art does not teach or suggest, either alone or in combination, a data storage system having a memory circuit board configured to, among other things, parse a payload to identify a series of individual instructions in response to a script command, and perform a series of operations in accordance with the identified series of individual instructions, as recited in claim 8. Rather, AAPA discloses a conventional data storage system where memory circuit boards of a data storage system require several separate communications (page 2, lines 24-25) resulting in poor response time due to delays, e.g., handshaking delays in waiting for a memory circuit board to complete an operation incrementally before sending a communication for a next operation, resource contention delays due to contention for a bus or buses, etc. (page 3, lines 3-8). Even in the Office Action states in paragraph 38 that AAPA does not teach receiving a communication that includes a script command and a payload, the payload including a series of individual instructions, and in response to the script command, parsing the payload to identify the series of individual instructions. Applicant agrees.

However, the Office Action then contends that Lemmon discloses in the second paragraph of page 9 that some amount of overhead is required for every data transfer that is required, and that a person of ordinary skill in the art would have recognized that by packaging multiple items (in this case instructions) into a single "packet" and then transmitting the packet, the overall number of data transfers will be reduced. Applicant respectfully disagrees with this contention. In particular, Applicant cannot find where Lemmon discloses packaging multiple items (in this case instructions) into a single "packet" and then transmitting the packet to reduce the overall number of data transfers. Applicant respectfully submits that this is a mischaracterization of the teachings of Lemmon. Lemmon discloses equalizing the bandwidth between buses without sacrificing any

bandwidth on the lower bandwidth bus and without introducing any buffering in a control logic device (see the Abstract, lines 3-6 of Lemmon).

Moreover, it is unclear how one could modify or why one would want to modify AAPA to use packets as suggested by the Office Action. The AAPA discloses a data storage system which uses one or more buses to interconnect circuit boards in the traditional sense (e.g., see page 3, lines 3-8 of the Specification). Nothing in any of the other references (e.g., Lemmon, Sharangpani, etc.) discloses how one could successfully make such a modification.

For the reasons stated above, claim 8 patentably distinguishes over the cited prior art, and the rejection of claim 8 under 35 U.S.C. §103(a) should be withdrawn. Accordingly, claim 8 is in allowable condition.

Because claims 9-14 and 25-26 depend from and further limit claim 8, claims 9-14 and 25-26 are in allowable condition for at least the same reasons.

Claims 15-21 and 27-28

Claim 15 is directed to a memory circuit board for a data storage system. The memory circuit board includes (a) an input/output port to connect with a processor circuit board of the data storage system; (b) a set of memory locations, at least some of the memory locations forming a cache that temporarily stores copies of data elements stored in a set of storage devices of the data storage system; and (c) a controller coupled to the input/output port and the set of memory locations. The controller is configured to (i) receive, from the processor circuit board through the input/output port, a communication that includes a script command and a payload, the payload including a series of individual instructions, (ii) parse the payload to identify the series of individual instructions in response to the script command, and (iii) perform a series of operations in accordance with the identified series of individual instructions.

The cited prior art does not teach or suggest, either alone or in combination, a memory circuit board for a data storage system where the

memory circuit board is configured to, among other things, parse a payload to identify a series of individual instructions in response to a script command, and perform a series of operations in accordance with the identified series of individual instructions, as recited in claim 15. Rather, as mentioned above in connection with claim 8, AAPA does not disclose such any mechanism that parses a payload to identify a series of individual instructions in response to a script command, and performs a series of operations in accordance with the identified series of individual instructions. Moreover, the Office Action even states that AAPA does not teach receiving a communication that includes a script command and a payload, the payload including a series of individual instructions, and in response to the script command, parsing the payload to identify the series of individual instructions, and Applicant agrees. Furthermore, nothing in Lemmon teaches or suggests how one could modify AAPA, as also explained above in connection with claim 8.

For the reasons stated above, claim 15 patentably distinguishes over the cited prior art, and the rejection of claim 15 under 35 U.S.C. §103(a) should be withdrawn. Accordingly, claim 15 is in allowable condition.

Because claims 16-21 and 27-28 depend from and further limit claim 15, claims 16-21 and 27-28 are in allowable condition for at least the same reasons.

Claims 22 and 29-30

Claim 22 is directed to a processor circuit board for a data storage system. The processor circuit board includes (a) an input/output port to connect with a memory circuit board of the data storage system; and (b) control circuitry coupled to the input/output port. The control circuitry is configured to provide, to the memory circuit board through the input/output port, a communication that includes a script command and a payload. The payload includes a series of individual instructions. The script command is configured to direct the memory circuit board to (i) parse the payload to identify the series of individual

instructions in response to the script command, and (ii) perform a series of operations in accordance with the identified series of individual instructions.

The cited prior art does not teach or suggest, either alone or in combination, a processor circuit board for a data storage system where the processor circuit board is configured to, among other things, provide a communication, which includes a script command and a payload, to a memory circuit board to (i) parse the payload to identify a series of individual instructions in response to the script command, and (ii) perform a series of operations in accordance with the identified series of individual instructions, as recited in claim 22. Rather, as mentioned above in connection with claim 8, AAPA does not disclose such any mechanism that parses a payload to identify a series of individual instructions in response to a script command, and performs a series of operations in accordance with the identified series of individual instructions. Moreover, the Office Action even states that AAPA does not teach receiving a communication that includes a script command and a payload, the payload including a series of individual instructions, and in response to the script command, parsing the payload to identify the series of individual instructions, and Applicant agrees. Furthermore, nothing in Lemmon teaches or suggests how one could modify AAPA, as also explained above in connection with claim 8.

For the reasons stated above, claim 22 patentably distinguishes over the cited prior art, and the rejection of claim 22 under 35 U.S.C. §103(a) should be withdrawn. Accordingly, claim 22 is in allowable condition.

Because claims 29-30 depend from and further limit claim 22, claims 29-30 are in allowable condition for at least the same reasons.

Conclusion

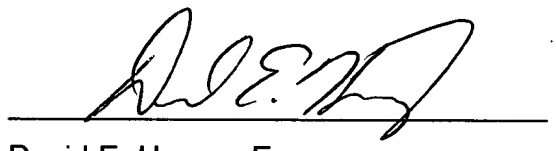
In view of the foregoing remarks, this Application should be in condition for allowance. A Notice to this affect is respectfully requested. If the Examiner believes, after this Amendment, that the Application is not in condition for

allowance, the Examiner is respectfully requested to call the Applicant's Representative at the number below.

Applicant hereby petitions for any extension of time which is required to maintain the pendency of this case. If there is a fee occasioned by this Amendment, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 50-0901.

If the enclosed papers or fees are considered incomplete, the Patent Office is respectfully requested to contact the undersigned collect at (508) 366-9600, in Westborough, Massachusetts.

Respectfully submitted,



David E. Huang, Esq.
Attorney for Applicant
Registration No.: 39,229
CHAPIN & HUANG, L.L.C.
Westborough Office Park
1700 West Park Drive
Westborough, Massachusetts 01581
Telephone: (508) 366-9600
Facsimile: (508) 616-9805

Attorney Docket No.: EMC01-31(01129)

Dated: May 20, 2005